UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/704,916	11/02/2000	William Brewster Robinson	99-969	4616

32127 7590 10/19/2004

VERIZON CORPORATE SERVICES GROUP INC.
C/O CHRISTIAN R. ANDERSEN
600 HIDDEN RIDGE DRIVE
MAILCODE HQE03H14
IRVING, TX 75038

EXAMINER	
HECK, MICHAEL C	

ART UNIT	PAPER NUMBER
3623	

DATE MAILED: 10/19/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/704,916	ROBINSON ET AL.	
	Examiner	Art Unit	
	Michael C. Heck	3623	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 July 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-72* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-72* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *01 July 2004* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This Final Office Action is responsive to applicant's amendment filed 1 July 2004. Applicant's amendment of 1 July 2004 amended claims 1, 3, 5, 7, 17, 19, 21, 31, 33-35, 37, 40, 49-51, 53, 56, and 65, and added new claims 71 and 72. Currently, claims 1-72 are pending.

Response to Amendment

2. The objection to the drawings in the First Office Action is not withdrawn in response to the applicant's amendment to the specification and submission of corrected drawings. However, item 2 of the First Office Action concerning reference character "58" and item 3 concerning reference sign "222" have been corrected. Item 3, reference sign "72", has not been corrected nor all of items 4 and 5. The corrected drawings submitted with the amendment filed 1 July 2004 were the same as the original drawings and did not reflect the corrections required in reply to the First Office Action.

3. The objection to the specification in the First Office Action is withdrawn in response to the applicant's amendment to the specification.

Response to Arguments

4. Applicant's arguments with respect to claims 31 and 1; 3, 17, 33, and 49; 4, 18, 34, and 50; 5,19, 35, and 51; 7, 21, 37, and 53; 65; and new claims 71 and 72 have been considered but are moot in view of the new ground(s) of rejection. All the said claims were amended except claims 4 and 18. The examiner notes that the applicant's

argument with regard to claims 4, 18, 34, and 50; 5, 19, 35, and 51; and 65 was incomplete. Applicant asserts the cited references do not teach the limitation of the claims, however no reason is given to support the applicant's assertion.

5. Applicant's arguments with respect to claims 15, 29, 47, and 63 have been fully considered but they are not persuasive. Applicant argues that other than the notification page, the office action fails to cite any support for the pages in the group as claimed. The applicant indicates that claim 47, as well as claims 15, 29, and 63 recite the limitations of the control page comprising a page selected from a group consisting of a worklist page, a process instance page, a process template page, a work item control page, a filter control page, and a notification page. In response, the examiner interpreted, based on the wording of the claim, that demonstrating art for one of the listed limitations is sufficient. Please see the 35 U.S.C. 103(a) rejection below.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. **Claims 1, 2, 4, 7-16, 18, 21-32, 34, 37-48, 50, and 53-71** are rejected under 35 U.S.C. 103(a) as being unpatentable over Brandt et al. (U.S. Patent 5,892,905) in view of Click, Jr. et al. (U.S. Patent 6,523,570). Brandt et al. disclose a flexible web-based interface for workflow management systems comprising:

- **[Claim 31]** configuring a set of predefined protocol user interface pages comprising at least one control page with said predefined protocol having at least one server-side script embedded therein (col. 14, lines 43-60, and col. 15, lines 64-66, Brandt et al. teach an interface components mechanism that uses HTML variables and templates. A user performs an action that causes the web browser to request access to a software application via the WWW by inputting data to a web server application. The input data comprises an URL or other address data that specifies the location of a HTML template. The HTML templates include input variables that are used to pass data between the web browser and the software application.);

- calling at least one server program with the at least one control page which thereby invokes at least one of the first workflow platform-dependent object and the at least one second workflow platform-independent object, wherein, when the functionality of the workflow management system is to be accessed via the set of predefined protocol user interface pages, the at least one control page calls the at least one server program which, in turn, invokes at least one of the first and second objects to promote data translation and exchange between the client program and the workflow management system (col. 19, lines 10-41, col. 19, line 62 to col. 20, line 3, col. 20, line 54-63, Brandt et al. teach FlowMark as a popular process engineering tool that allows a relatively complex project or task to be broken down into a series of smaller processes or tasks. Information is processed by the FlowMark workflow software and usually involves multiple related activities. The Internet/application gateway includes a Common Gateway Interface (CGI), a FlowMark/Internet Gateway (FMIG), and WWW Application Program Interfaces (APIs). The user who needs to access a FlowMark application over the WWW will input a request to a web browser using a client workstation. The user can enter a URL for a specific home page site or click on a button presented in an HTML-generated user interface using the web browser. When the user "submits" the requested information, usually by clicking on a button on an HTML form, a web server application receives the input data from the web browser. After receiving the data from the web browser, the CGI parses the data to locate relevant information about the requested processes, including the request for access to FlowMark. The CGI sends the user data and requests to the FMIG along with some control information. The FMIG provides a way for FlowMark applications to interact with a web user over the WWW. The FMIG directs the flow of information between the CGI and the FlowMark application and initiates FlowMark functions by using FlowMark APIs.).

Brandt et al. does not teach pointing at least one first workflow platform-dependent object to access the workflow functionality, wherein the at least one first workflow

platform-dependent object is customized for the workflow management system, and interfacing at least one second workflow platform-independent object with the at least one first workflow platform dependent object, wherein the at least one second workflow platform independent object is configured to provide input data received from the client server to the at least one first workflow platform dependent object and to receive output data provided by the workflow management system from the at least one first workflow platform dependent object. However, Brandt et al. does teach that a user who needs to access a FlowMark application over the WWW will input a request to a web browser using a client workstation. FlowMark is a workflow application (col. 19, lines 2-4 and 52-64). Clicks, Jr. et al. teach porting software between different computing platforms. A platform specific compiler includes platform dependent compiler object code and platform independent compiler object code, which are suitable for execution on a particular hardware platform. An interface that is partially embedded in the platform independent object code and partially embedded in the platform dependent object code mediates flow of information between the platform independent compiler code and the platform dependent compiler object code during platform specific compiler run time. During run time (execution), the platform independent compiler object code, interacts with the platform dependent compiler object code to operate (i.e., compiler) in a target dependent manner (col. 1, lines 36-39, col. 2, lines 42-51, col. 6, lines 16-22). It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to use the platform specific compiler of Click, Jr. et al. with the teachings of Brandt et al. since Brandt et al. teach the application gateway facilitates a response to

the request by formatting the appropriate commands to the software application (col. 9, lines 8-34). Providing a common user interface for accessing various software applications over the WWW allows for increased productivity with greater efficiency (Brandt et al.: col. 33, lines 32-40). The multi-platform compiler system in conjunction with the architecture design file particular to a specific target platform substantially reduces the time required to port a compiler from one platform to another different platform by substantially reducing the use of manual code generation (Click, Jr. et al.: col. 4, lines 46-53). The success or failure of an enterprise depends to a large extent on the quality of the decision making within the enterprise. The domain or the "extent of the world" used to make the decision leads to more optimal decision. Having access to different application programs via the World Wide Web through a common user interface expands the domain of the decision making process. Therefore, since the "extent of the world" is increased, the decisions made will be more optimal.

- **[Claim 32]** the step of logging a user on to the workflow management system via one of the set of predefined protocol pages, receiving a user identification variable and a password variable therefrom, and invoking at least one of the first and second objects to authenticate the user identification variable with the workflow management system (Brandt et al.: col. 8, lines 30-39, Brandt et al. teach that web security uses a password and userID combination to authenticate a particular web user to access a particular web server or specific resource through the web server).

- [**Claim 34**] the step of issuing a redirection command to cause a predefined protocol page represented by the target user interface address to be presented to the user for performing work on the task identified by the work item identification (Brandt et al.: col. 15, lines 25-44, Brandt et al. teach the supplied HTML templates include one or more variables. The values for the variables are requested from the associated software application. The software application either receives the date corresponding to the variable or initiates software processes to generate the appropriate data.).

- [**Claim 37**] the step of a receiving predefined process template identification from the at least one control page, and invoking at least one of the first and second objects to initiate a new instance of a process template in the workflow management system corresponding to the process template identification, wherein the new instance of the process template defines a specific set of tasks (Brandt et al.: col. 15, lines 3-23, Brandt et al. teach that in processing the request for input data, the software application may initiate additional software processes to generate the data, call other programs that have the data, or retrieve data for local and/or networked data storage. The software application then returns the requested data to the gateway. The gateway then substitutes the variables in the TML template with the data retrieved from the software application. The gateway then outputs the HTML template to the web server with the real data substituted for the substitution variables. To facilitate the use of the Internet/application gateway as a gateway between the software application and the web server, a library of HTML templates are provided. The library of HTML templates provides a flexible and easily customizable way of providing access to multiple software applications.).

- [**Claim 38**] the step of determining whether input data is needed to initiate the new instance of the predefined process template (Brandt et al.: col. 16, lines 7-23, Brandt et al. teach an input variables are inserted into HTML pages to provide input from web browsers to the Internet/application gateway and software applications. The Internet/application gateway can be configured to pass a particular variable from one HTML screen to the next HTML screen. The examiner interprets to process to include a determination since a particular variable is to be identified.).

- [**Claim 39**] the step of redirecting the user to a user interface page to receive required input data to properly initiate the new instance of the process template with the input data if the determining step results in a determination that data is needed to initiate the process instance (Brandt et al.: col. 16, lines 7-46, Brandt et al. teach input variables are inserted into HTML pages to provide input from web browsers to the Internet/application gateway and software applications. The Internet/application gateway can be configured to pass a particular variable from one HTML screen to the next HTML screen. User-defined variables can be added to HTML templates by system operators to provide specialized inputs and outputs as needed for different types of software applications. The examiner interprets the process to include a determination since a particular variable is to be identified.).

- [**Claim 40**] the step of redirecting the user from the user interface page back to the New Instance servlet to update the workflow management system with the received input data to initiate the process instance (Brandt et al.: col. 15,

lines 25-44, and col. 16, lines 7-46, Brandt et al. teach software applications receive data corresponding to the variables or initiate software processes to generate the appropriate data. Input variables are inserted into HTML pages to provide input from web browsers to the Internet/application gateway and software applications. The Internet/application gateway can be configured to pass a particular variable from one HTML screen to the next HTML screen.).

- [**Claim 41**] at least one predefined protocol user interface page adapted to receive at least one data variable from the user and to call the at least one server-based applet (Brandt et al.: col. 15, lines 64-66, Brandt et al. teach the HTML templates include input variables that are used to pass data between the web browser and the software application).

- [**Claim 42**] the user interface page further comprises at least one <FORM> tag having at least one input element for receiving data from the user (Brandt et al.: figure 11-22, Brandt et al. teach <FORM ACTION> is used.)

- [**Claim 43**] the user interface page is adapted to provide at least one data variable to initiate a process template into a running process that requires the data entry for instantiation (Brandt et al.: figure 11-22, and col. 23, lines 5-41, Brandt et al. teach an INPUT TYPE is needed, such as compact, mid size, full size or luxury for the rental reservation example. An activity program receives car rental information, locates the next reservation number, saves the reservation number to a file, returns the reservation number to the requester and sets the reservation number in the output data container.).

- [**Claim 44**] the user interface page is adapted to provide at least one data variable to complete a task from a previously initiated process that requires the data entry for completion (Brandt et al.: figure 11-22, and col. 23, lines 5-41, Brandt et al. teach an activity program receives car rental information, locates the next reservation number, saves the reservation number to a file, returns the reservation number to the requester and sets the reservation number in the output data container. Figure 11 data input is used for figure 13 and figure 14.).

- [**Claim 45**] the first and second objects are Java classes (Brandt et al.: col. 6, lines 4-12, Brandt et al. teach other types of data besides HTML may be used to be transmitted to a web browser including Java applets (executable code)).

- [**Claim 46**] the Java class comprises a Java interface class (Brandt et al.: col. 6, lines 4-12, Brandt et al. teach other types of data besides HTML may be used to be transmitted to a web browser including Java applets (executable code)).

- [**Claim 47**] the at least one control page comprises a page selected from a group consisting of a worklist page, a process instance page, a process template page, a work item control page, a filter control page, and a notifications page (Brandt et al.: col. 28, lines 9-56, Brandt et al. teach the process will send a reservation confirmation HTML screen to the web client at the clients workstation. Figure 13 shows the HTML code for the reservation confirmation template. The examiner interprets the reservation confirmation template to be the notification page.).

- [**Claim 48**] the client program is a web browser and the client server is a web server (Brandt et al.: col. 5, lines 52-54, and col. 6, lines 3-5, Brandt et al. teach the web browser is a software program running on the clients workstation and the client workstation and web server computer system are the same physical and/or logical computer system.).

- **[Claim 72]** directing the user to a user interface filter page; receiving filter parameters via the user interface filter page; invoking at least one of the first and second objects to filter a work list with the workflow management system (Brandt et al.: col. 15, lines 3-23, Brandt et al. teach that in processing the request for input data, the software application may initiate additional software processes to generate the data, call other programs that have the data, or retrieve data for local and/or networked data storage. The software application then returns the requested data to the gateway. The gateway then substitutes the variables in the HTML template with the data retrieved from the software application. The gateway then outputs the HTML template to the web server with the real data substituted for the substitution variables. To facilitate the use of the Internet/application gateway as a gateway between the software application and the web server, a library of HTML templates are provided. The library of HTML templates provides a flexible and easily customizable way of providing access to multiple software applications.).

Claims 1, 2, 4, 7-16, 18, 21-30, 50, 53-64, and 71 substantially recites the same limitations as that of claims 31, 32, 34, 37-48, and 72 with the distinction of the recited method being an interface and method. Hence the same rejection for claims 31, 32, 34, 37-48, and 72 as applied above applies to claims 1, 2, 4, 7-16, 18, 21-30, 50, 53-64, and 71.

- [**Claim 65**] creating at least one predefined protocol process activity page relating to a process and named for the unique process identifier, wherein the

process corresponds with a function of the workflow management system (Brandt et al.: col. 7, lines 1-40, and col. 23, line 40 to col. 24, line 23, Brandt et al. teach when a web server application that is running on a web server computer receives a web page request from a web browser, it will build a web page in HTML or retrieve a file containing a pre-built web page and send it across a connection to the requesting browser. Some web pages are designed to elicit input from a web browser. Referring to the Car Rental Example, the person or user who wants to rent a car will access the WWW by using a client workstation, which is running a web browser. The user will enter the URL for the rental car agency and locate the home page site for the rental car agency using the web browser. The web server receives an input from a web browser specifying the HTML template of a rental reservation form as the next output that needs to be sent back to the web browser. Once the user has input the information, the user submits the information by clicking on a "submit" button on the rental reservation form. At this point, the web server application receives the data stream generated by the user request from the web browser. One suitable format picks out all variables and other relevant information data and sends it to a web server application in a post data stream format. The web server application examines the data stream from the web browser to determine what action should be taken to fulfill the user's request. The examiner interprets the Car rental example as a workflow management system and the HTML template of a rental reservation form is a process activity page relating to a process and named for the unique process identifier.);

- locating the at least one predefined protocol process activity page in the predefined interface root directory path (Brandt et al.: col. 23, lines 42-57, Brandt et al. teach the person or user who wants to rent a car will access the WWW by using a client workstation that is running a web browser and will enter the URL for the rental car agency and locate the home page site for the rental car agency. Click, Jr. et al. : col. 3, lines 18-30, Figures 6A and 6B, Click, Jr. et al. teach an apparatus for compiling a platform specific compiler that includes a set of user defined platform dependent compiler architecture descriptors that describe corresponding architectural features of a particular hardware platform. An architecture descriptor compiler converts the user defined platform dependent compiler architecture descriptors into the platform dependent compiler source code, which is converted into platform dependent object code by a host compiler. During run-time for the platform specific compiler, an interface mediates the flow of information between platform dependent compiler object code and platform independent compiler object code.);

- creating a process directory beneath the predefined interface root directory path for the process and named for the unique identifier thereof (Brandt et al.:

col. 26, lines 49-56, Brandt et al. teach that when the car rental reservation process model was initially built, the first activity program was identified and designated to run automatically whenever the car reservation process model was invoked. There will be multiple related activity programs that will work together to process the car rental request. Each individual activity program is a separate software module that is designated to accomplish a specific task or return some requested information. Click, Jr. et al. : col. 3, lines 18-30, Figures 6A and 6B, Click, Jr. et al. teach an apparatus for compiling a platform specific compiler that includes a set of user defined platform dependent compiler architecture descriptors that describe corresponding architectural features of a particular hardware platform.);

- creating at least one predefined protocol user interface page within the created process directory in a predetermined protocol relating to a task assignable within the process and named for the task unique identifier if the process requires input on any of its assignable activities (Brandt et al.: col. 23, lines 42-57, Brandt et al. teach the person or user who wants to rent a car will access the WWW by using a client workstation that is running a web browser and will enter the URL for the rental car agency and locate the home page site for the rental car agency. Click, Jr. et al. : col. 3, lines 18-30, Figures 6A and 6B, Click, Jr. et al. teach an apparatus for compiling a platform specific compiler that includes a set of user defined platform dependent compiler architecture descriptors that describe corresponding architectural features of a particular hardware platform. An architecture descriptor compiler converts the user defined platform dependent compiler architecture descriptors into the platform dependent compiler source code, which is converted into platform dependent object code by a host compiler. During run-time for the platform specific compiler, an interface mediates the flow of information between platform dependent compiler object code and platform independent compiler object code.);

- locating the at least one predefined protocol user interface page in the created directory within the predefined interface root directory path (Brandt et al.: col. 25, lines 21-33, and col. 26, lines 49-56, Brandt et al. teach that by using HTML templates with substitution variables, a single relatively simple GCI module in conjunction with an FMIG can provide an effective interface between a web server and a plurality of software applications. This allows system operators to provide easily customizable web access to a plurality of software applications over the WWW. When the car rental reservation process model was initially built, the first activity program was identified and designated to run automatically whenever the car reservation process model was invoked. There will be multiple related activity programs that will work together to process the car rental request. Each individual activity program is

a separate software module that is designated to accomplish a specific task or return some requested information.); and

- whereby the predefined protocol process activity page can be automatically located by the interface within the predefined interface root directory path of the client server by only knowing the process unique identifier and the at least one predefined protocol user interface page can be located in the created directory within the predefined interface root directory path by knowing only the task unique identifier (Brandt et al.: col. 26, lines 49-56, Brandt et al. teach that when the car rental reservation process model was initially built, the first activity program was identified and designated to run automatically whenever the car reservation process model was invoked. There will be multiple related activity programs that will work together to process the car rental request. Each individual activity program is a separate software module that is designated to accomplish a specific task or return some requested information.).

- [**Claim 66**] the step of embedding a form within the at least one user interface page in the predefined protocol configured so as to provide any required data to the assignable task into the workflow management system (Brandt et al.: col. 25, lines 21-33, and col. 26, lines 49-56, Brandt et al. teach that by using HTML templates with substitution variables, a single relatively simple GCI module in conjunction with an FMIG can provide an effective interface between a web server and a plurality of software applications. This allows system operators to provide easily customizable web access to a plurality of software applications over the WWW. When the car rental reservation process model was initially built, the first activity program was identified and designated to run automatically whenever the car reservation process model was invoked. There will be multiple related activity programs that will work together to process the car rental request. Each individual activity program is a separate software module that is designated to accomplish a specific task or return some requested information.).

- [**Claim 67**] the form contains input prompts configured so as to provide specific data in a machine readable format to the workflow management system (Brandt et al.: col. 23, line 58 to col. 24, line 23, Brandt et al. teach the user inputs information such as member number, last name, car preference and submits information by clicking on the "submit" button. The web server application examines the data stream to determine what action should be taken to fulfill the user's request.).

- [**Claim 68**] the step of embedding a hidden field on the at least one user interface page containing the unique process identifier for cross-referencing the data within the at least one user interface page with the workflow

management system (Brandt et al.: col. 26, lines 1-12, Brandt et al. teach that to assure that FMIG can match up the process instance with the web browser that requested it, the FMIG generates and stores a "handle" for the web client that is some combination of the process instance name, the activity instance name, and the security data for the web client. The examiner interprets the "handle is an embedded hidden field.).

- [**Claim 69**] the predefined protocol comprises at least one of HTML and javascript (Brandt et al.: col. 6, lines 13-21, Brandt et al. teach software programs running on a web server computer system typically output data pages of HTML data to web browsers in response to requests.).

- [**Claim 70**] the step of defining a programming object for use as an input container for delivering data entered by a user on the at least one user interface page (Brandt et al.: col. 7, lines 23-40, Brandt et al. teach a web page may request the user's name in an HTML form and require the user to select a particular function using an HTML button).

8. **Claims 3, 5, 17, 19, 33, 35, 49, and 51** are rejected under 35 U.S.C. 103(a) as being unpatentable over Brandt et al. (U.S. Patent 5,892,905) and Click, Jr. et al. (U.S. Patent 6,523,570) in view of Smith (Smith, Protocol Work Melds Storage Methods, Network World, 13 September 1999, p. 67 [PROQUEST]). As to claim 33, Brandt et al. and Click, Jr. et al. disclose a flexible web-based interface for workflow management systems comprising the step of receiving a work item identification and a target user interface address from the at least one control page. Brandt et al. teach a user performs an action that causes the web browser to request access to a software application via the WWW by inputting data to a web server application. The input data comprises an URL or other address data that specifies the location of an HTML template (Brandt et al.: col. 14, lines 43-60). Brandt et al. and Click, Jr. et al. fail to teach invoking at least one of the first and second objects to record lock a task in the

workflow management system corresponding to the work item identification and permit exclusive access by a user to the task identified by the work item identification by checking out the task to the user for the user to work on the task. Smith teaches that as for dealing with instances in which more than one user wants access to the same data simultaneously, vendors can use the Common Internet File Services (CIFS) protocol's soft locking mechanism to manage file system coherency and deal with multi-user contention. The soft lock operates as a file or record lock that can be relinquished for contention-resolution purposes (Para 9). It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to use the CIFS protocol's soft locking mechanism of Smith with the teachings of Brandt et al. and Click, Jr. et al. since Brandt et al. teach multiple users connected to the system (Brandt et al.: col. 10, lines 42-50). Providing a common user interface for accessing various software applications over the WWW allows for increased productivity with greater efficiency (Brandt et al.: col. 33, lines 32-40). Extending these file-sharing protocols will let vendors deliver storage-area network (SAN) that can share data with any authorized client on the network (Para 11). The success or failure of an enterprise depends to a large extent on the quality of the decision making within the enterprise. The domain or the "extent of the world" used to make the decision leads to more optimal decision. Having access to different application programs and data via the World Wide Web through a common user interface expands the domain of the decision making process. Therefore, since the "extent of the world" is increased, the decisions made will be more optimal.

- **[Claim 35]** the step of receiving a work item identification from the at least one control page and invoking at least one of the first and second objects to release a record lock on a task in the workflow management system corresponding to the work item identification and terminate any exclusive access by a user to work on the task identified by the work item identification (Brandt et al.: col. 16, line 49 to col. 17, line 25, Brandt et al. teach an identifier is created to identify the specific communication between the user and the software application. The identifier is attached to all information transmitted between the user and the software application. Smith: Para 9, Smith teaches that as for dealing with instances in which more than one user wants access to the same data simultaneously, vendors can use the Common Internet File Services (CIFS) protocol's soft locking mechanism to manage file system coherency and deal with multi-user contention. The soft lock operates as a file or record lock that can be relinquished for contention-resolution purposes. For example, if one client wished to write to the same file as a second client, the NAS/SAN server revokes the soft lock given to the first client and takes over the management of write operations using traditional NAS methodologies.).

Claims 3, 5, 17, 19, 49, and 51 substantially recites the same limitations as that of claims 33 and 35 with the distinction of the recited method being an interface and method. Hence the same rejection for claims 33 and 35 as applied above applies to claims 3, 5, 17, 19, 49, and 51.

9. **Claims 6, 20, 36 and 52** are rejected under 35 U.S.C. 103(a) as being unpatentable over Brandt et al. (U.S. Patent 5,892,905) and Click, Jr. et al. (U.S. Patent 6,523,570) in view of Notani et al. (U.S. Patent 6,397,191). As to claim 36, Brandt et al. and Click, Jr. et al. disclose a flexible web-based interface for workflow management systems but fail to teach the step of receiving task-specific data from a requesting page and updating the task identified by the work item identification with the task-specific data in the workflow management system. Notani et al. teach the process for data access and transformation for an object-oriented workflow includes supporting communication

of objects and derived format objects built from native format objects. The process involves communicating information between activities of an executing workflow using objects and derived format objects (Notani et al.: col. 2, lines 40-51). The examiner interprets activities as tasks. It would have been obvious to one of ordinary skill in the art at the time of the applicant's invention to include the process for data access and transformation of Notani et al. with the teachings of Brandt et al. and Click, Jr. et al. since Brandt et al. teaches providing standard procedures, routines, tools, and software "hooks" for accessing software applications over the WWW (col. 3, lines 56-65). Maximizing productivity and efficiency while maintaining flexibility and minimizing cost enhances a company's competitiveness in the marketplace. Having access to the collaborative management systems from remote locations via the Internet allows managers to be continually engaged in the management of the company and their external relationships. Enterprise collaboration provides an advantage over conventional supply chain, enterprise and site planning environments (Notani et al.: col. 2, lines 21-24). Providing a common user interface for accessing various software applications over the WWW allows for increased productivity with greater efficiency (Brandt et al.: col. 33, lines 32-40). As such, productivity and efficiency of the manger are enhanced and the flexibility of the manager is greatly increased. Cost to implement is minimal since access to the Internet is fairly common wherever the manager may be. Therefore, access to the company and collaborative management systems from remote locations is a great advantage for the manager and company and helps the company maintain it competitiveness in the marketplace.

Claims 6, 20 and 52 substantially recites the same limitations as that of claim 36 with the distinction of the recited method being an interface and method. Hence the same rejection for claim 36 as applied above applies to claims 6, 20 and 52.

Conclusion

10. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael C. Heck whose telephone number is (703) 305-8215. The examiner can normally be reached Monday thru Friday between the hours of 8:00am - 4:30pm. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tariq R. Hafiz can be reached on (703) 305-9643. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 308-1113.

Any response to this action should be mailed to:

Director of the United States Patent and Trademark Office
P.O. Box 1450
Alexandria, Virginia 22313-1450

Or faxed to:

(703) 872-9306 [Official communications; including After Final communications labeled "**Box AF**"]

(703) 746-9419 [Informal/Draft communication, labeled "**PROPOSED**" or "**DRAFT**"]

Hand delivered responses should be brought to 220 South 20th Street, Crystal Plaza Two, Lobby, Room 1B03, Arlington, Virginia 22202.

mch
13 October 2004

TARIQ R. HAFIZ
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 3600